Test Plan

Created By: Shreyas Bhagat

Date: 26/06/2024

Table of Contents

Overview

As part of the project, 'OpenCart' asked Shreyas to test a few functionalities of the *"https://demo.opencart.com/"* web application. This document serves as a high-level test planning document with details on the scope of the project, test strategy, test schedule, resource requirements, test deliverables, and schedule.

Scope of the Project

The scope of this project includes testing the core functionalities of the OpenCart demo web application to ensure its reliability, performance, and user experience. The primary focus will be on the following areas:

Inclusions

- Register
- Login & Logout
- Forgot Password
- Search
- Product Compare
- Product Display Page
- Add to Cart
- Wish List
- Shopping Cart
- Currencies
- Home Page
- Checkout Page
- My Account Page
- Order History Page
- Downloads Page
- Contact Us Page
- Menu Options
- Footer Options
- Category Pages

Test Environments

Testing will be conducted across various environments to ensure compatibility and performance:

- **Windows 10**: Chrome, Firefox, and Edge
- **Mac OS**: Safari
- **Devices: Desktop, Computers, Laptops, and Smartphones, etc**
- **Android Mobile OS: Chrome**
- **iPhone Mobile OS: Safari**

Environment Name	*Environment URL*
QA	http://localhost/opencart/
Prod	https://demo.opencart.com/

Test Strategy

The test strategy will involve the following approaches:

Step #1: Creation of Test Scenarios and Test Cases

- Test Designing Techniques:
  - Equivalence Class Partition
  - Boundary Value Analysis
  - Decision Table Testing
  - State Transition Testing
  - Use Case Testing
- Expert Techniques:
  - Error Guessing
  - Exploratory Testing
- Prioritization:
  - Test cases will be prioritized based on functionality and criticality.

Step #2: Testing Process

- Initial Testing:
  - Perform Smoke Testing to check the core functionalities of the application.
  - Reject the build if Smoke Testing fails, and wait for a stable build before proceeding.
- In-depth Testing:
  - Upon receiving a stable build, conduct in-depth testing using the created test cases.

- Multiple test resources will test the application on various supported environments simultaneously.

- Defect Reporting:
  - Report bugs using a bug-tracking tool.
  - Send daily status emails to development management with the defects found.
- Types of Testing:
  - Smoke Testing and Sanity Testing
  - Regression Testing and Retesting
  - Usability Testing, Functionality, and UI Testing
- Test Cycles:
  - Repeat test cycles until a quality product is achieved.

Step #3: Best Practices

- Context-Driven Testing:
  - Perform testing based on the specific context of the application.
- Shift Left Testing:
  - Begin testing at the early stages of development rather than waiting for a stable build.
- Exploratory Testing:
  - Use expertise to conduct exploratory testing in addition to executing the test cases.
- End-to-End Flow Testing

Defect Reporting Procedure

During Test Execution:

- **Observation and Reporting**:
  - Any deviation from the expected behavior will be noted. If it cannot be reported as a defect, it will be logged as an observation/issue or posed as a question.
  - Usability issues will also be reported.
- **Reproducibility**:
  - After discovering a defect, it will be retested to verify its reproducibility. Screenshots and steps to reproduce will be documented.

- **Daily Reporting**:
    - At the end of each day, defects encountered will be sent along with observations.

Note:

- Defects will be documented in Excel.
- Test scenarios and Test cases will be documented in an Excel document.

Test Schedule

Phase	Start Date	End Date
Test Planning	2024-07-01	2024-07-07
Test Case Design	2024-07-08	2024-07-14
Test Environment Setup	2024-07-15	2024-07-17
Test Execution	2024-07-18	2024-07-31
Defect Reporting and Retesting	2024-08-01	2024-08-07
Test Closure	2024-08-08	2024-08-10

Resource Requirements

- **Testers**: 2 Manual Testers, 1 Automation Tester
- **Test Tools**: JMeter, Postman
- **Hardware**: Test servers, PCs/laptops with required configurations
- **Other Resources**: Access to the OpenCart demo application, test data

Roles/Responsibilities

Name	Role	Responsibilities
Person A	Test Manager	✓ Escalations

Person B	Test Lead	✓ Create the Test Plan and get the client signoffs ✓ Interact with the application, create and execute the test cases ✓ Report defects ✓ Coordinate the test execution. Verify the validity of the defects being reported. ✓ Submit daily issue updates and summary defect reports to the client. ✓ Attend any meeting with client.
Person C	Senior Test Engineer	✓ Interact with the application ✓ Create and Execute the Test cases. ✓ Report defects
Person D	Test Engineer	✓ Interact with the application ✓ Execute the Test cases. ✓ Report defects

Entry and Exit Criteria

Below are the entry and exit criteria for every phase of Software Testing Life Cycle:

Requirement Analysis

Entry Criteria:

- Receipt of Requirements Documents or project details by the testing team.

Exit Criteria:

- The list of requirements has been explored and understood by the testing team.
- All doubts and queries have been clarified.

Test Planning

Entry Criteria:

- Testable requirements derived from the Requirements Documents or project details.
- All doubts and queries have been clarified.

Exit Criteria:

- The Test Plan document (including the Test Strategy) is reviewed and signed off by the client.

Test Designing

Entry Criteria:

- Test Plan document is reviewed and signed off by the client.

Exit Criteria:

- Test Scenarios and Test Cases documents are reviewed and signed off by the client.

Test Execution

Entry Criteria:

- Test Scenarios and Test Cases documents are reviewed and signed off by the client.
- Application is ready for testing.

Exit Criteria:

- Test Case Reports and Defect Reports are prepared.

Test Closure

Entry Criteria:

- Test Case Reports and Defect Reports are prepared.

Exit Criteria:

- Test Summary Reports are completed.

Risks and Mitigations

The following is the list of risks possible and the ways to mitigate them: Risk: Non-Availability of a Resource

1. Non-Availability of a Resource
   - Mitigation: Backup Resource Planning
2. The Build URL is Not Working
   - Mitigation: Resources will work on other tasks
3. Less Time for Testing
   - Mitigation: Ramp up the resources based on the client's needs dynamically
4. Environment Issues
   - Mitigation: Maintain a backup environment and ensure regular checks and maintenance.
5. Unclear Requirements
   - Mitigation: Regular communication with the client to clarify requirements and obtain detailed documentation.
6. Defect Leakage to Production
   - Mitigation: Perform thorough regression testing and conduct final acceptance testing before release.

7. Test Data Issues
    - Mitigation: Create and maintain a comprehensive test data set and refresh it as needed.
8. Third-Party Integration Issues
    - Mitigation: Perform early and continuous integration testing with third-party services to identify and resolve issues promptly.

Approvals

The team will send different types of documents for Client Approval like below:

- Test Plan
- Test Scenarios
- Test Cases
- Reports

Testing will only continue to the next steps once these approvals are done.